

Contorni Attivi

Guido Cocchi, Francesco Morelli

Università degli Studi di Firenze

28 Maggio 2013

Indice I

- 1 Introduzione
- 2 Preliminari Matematici
- 3 Approcci trattati
 - Rappresentazione esplicita - Kass
 - Level Set Method
- 4 Chan Vese
 - Definizione del modello
 - Implementazione
 - Differenze Finite
 - Regola di aggiornamento
 - Algoritmo
- 5 Inizializzazione
 - Alcuni Esempi su immagini biomediche
- 6 Chan Vese Vector Image [CV2002]

Indice II

7 Chan Vese Multiphase [CS]

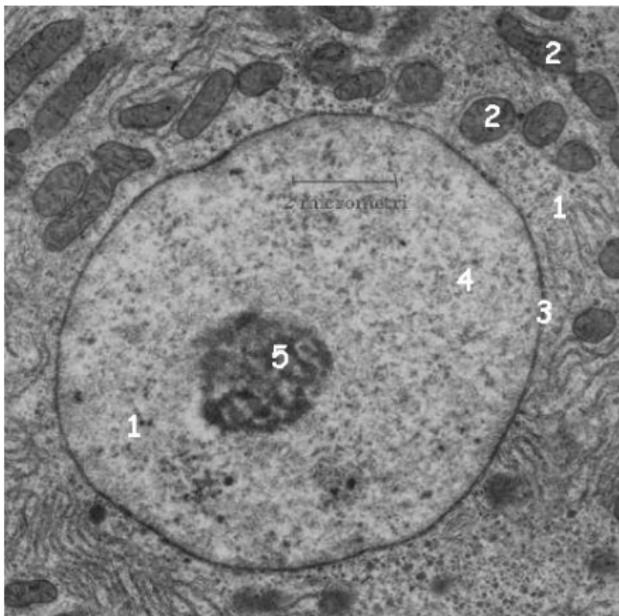
8 Sperimentazione

- Parametro: λ_1
- Parametro: λ_2
- Parametro μ
- Parametro ν
- Conclusioni sui parametri

9 Bibliografia

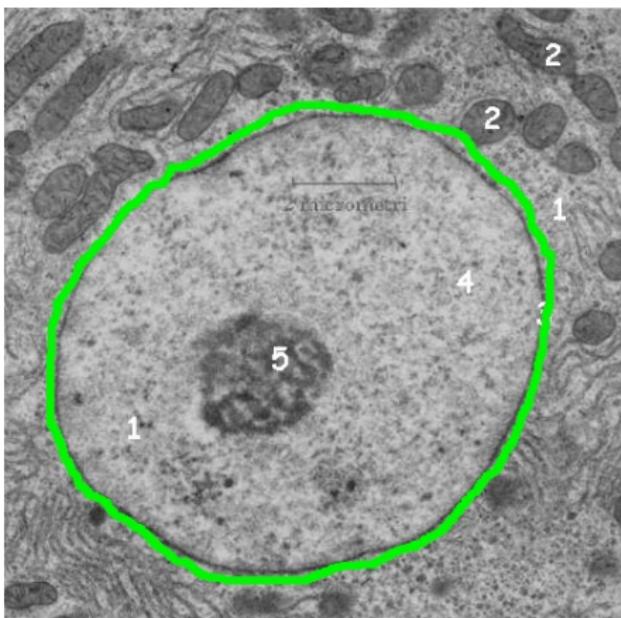
10 Sessione Laboratorio

Un problema ricorrente nel campo dell'analisi delle immagini riguarda la segmentazione: data un'immagine, vogliamo cercare di individuare le diverse regioni e oggetti che la compongono.





Un problema ricorrente nel campo dell'analisi delle immagini riguarda la segmentazione: data un'immagine, vogliamo cercare di individuare le diverse regioni e oggetti che la compongono.



Per prima cosa, vorrei distinguere le regioni di background e di foreground.
Per farlo potremmo immaginare una curva che piano piano si adatta ai contorni dell'oggetto in foreground delimitando così le due regioni.

Problemi

- Se volessi formalizzare matematicamente questa curva, quali parametri dovrei tenere in considerazione (lunghezza, regolarità ecc..)?

Per prima cosa, vorrei distinguere le regioni di background e di foreground.
Per farlo potremmo immaginare una curva che piano piano si adatta ai contorni dell'oggetto in foreground delimitando così le due regioni.

Problemi

- Se volessi formalizzare matematicamente questa curva, quali parametri dovrei tenere in considerazione (lunghezza, regolarità ecc..)?
- Secondo quale criterio potrei far evolvere la curva?
- Vedremo due possibili approcci per definire una curva: forma parametrica (Kass) e forma implicita(Chan Vese).

oooooo oooooooooooooooo

oo
ooo
ooo
o

Sia $C : [0, 1] \rightarrow \mathbb{R}^2$ una curva di classe $C^2(0, 1)$.

Sia $C : [0, 1] \rightarrow \mathbb{R}^2$ una curva di classe $C^2(0, 1)$.

Nella formulazione classica di Kass [KW], C evolve sul piano xy cercando di minimizzare il seguente funzionale:

$$\int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds \quad (1)$$

Sia $C : [0, 1] \rightarrow \mathbb{R}^2$ una curva di classe $C^2(0, 1)$.

Nella formulazione classica di Kass [KW], C evolve sul piano xy cercando di minimizzare il seguente funzionale:

$$\int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds \quad (1)$$

con:

- E_{interna} controlla la regolarità della curva

Sia $C : [0, 1] \rightarrow \mathbb{R}^2$ una curva di classe $C^2(0, 1)$.

Nella formulazione classica di Kass [KW], C evolve sul piano xy cercando di minimizzare il seguente funzionale:

$$\int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds \quad (1)$$

con:

- E_{interna} controlla la regolarità della curva
- E_{esterna} fa evolvere la curva in base alle caratteristiche dell'immagine.

Cos'è un contorno attivo?

Informalmente un contorno attivo :

- è una curva che evolve sul piano xy.

oooooo ooooooooooooooooooooo

oo
ooo
ooo
o

Cos'è un contorno attivo?

Informalmente un contorno attivo :

- è una curva che evolve sul piano xy.
- ha l'obiettivo di delineare i contorni dell'immagine minimizzando un funzionale di energia.

oooooo

oooooo

oooooooooooooooooooo

Vector

Multiphase

Sperimentazione

Bibliografia

Lab.

oo
ooo
ooo
o

Il problema della curva più corta: un esempio grafico

Il problema:

Dati un punto materiale A e un punto B nel piano xy , si vuole determinare la curva γ più corta che connette A e B.

ooooo

ooooo

oooooooooooooooooooo

Vector

Multiphase

Sperimentazione

Bibliografia

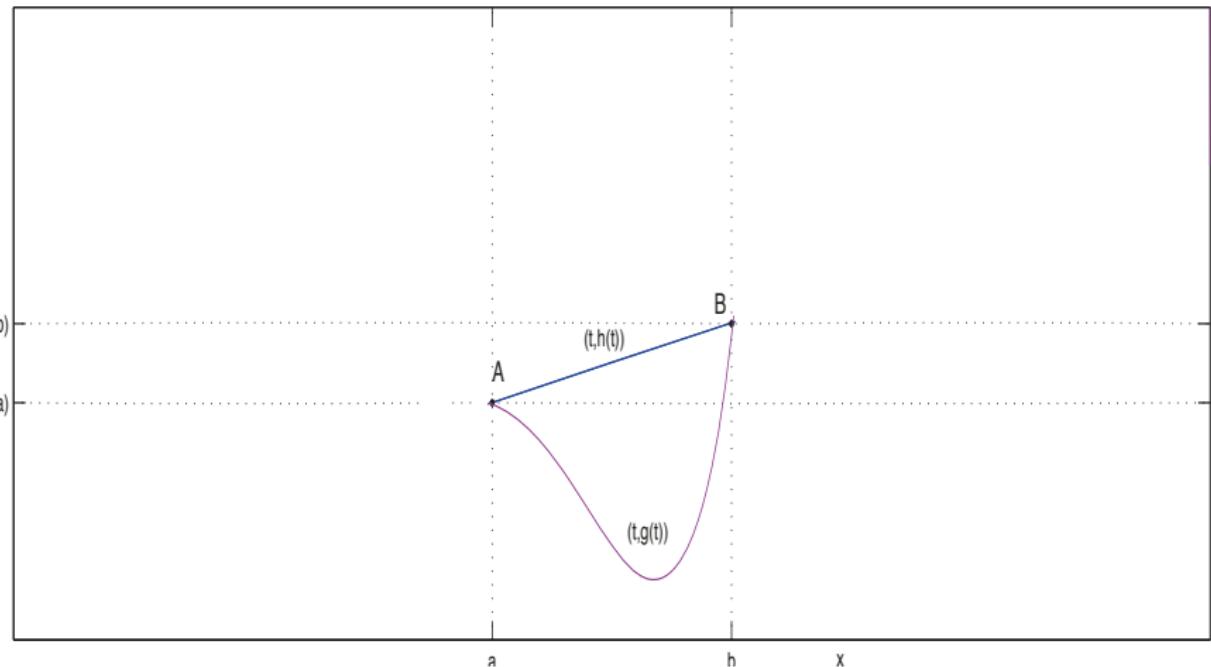
Lab.

oo
ooo
ooo
o

Il problema della curva più corta: un esempio grafico

Il problema:

Dati un punto materiale A e un punto B nel piano xy , si vuole determinare la curva γ più corta che connette A e B.



Definizione dell'universo di curve

$\mathcal{U} = \{\gamma \in C^1(a, b) | \gamma(a) = A, \gamma(b) = B\}$ è l'universo di tutte le possibili curve di interesse.

Definizione dell'universo di curve

$\mathcal{U} = \{\gamma \in C^1(a, b) | \gamma(a) = A, \gamma(b) = B\}$ è l'universo di tutte le possibili curve di interesse.

Si noti in particolare che una curva $\gamma(t) \in \mathcal{U}$ è del tipo $\gamma(t) = (t, g(t))$ con $t \in [a, b]$

Definizione dell'universo di curve

$\mathcal{U} = \{\gamma \in C^1(a, b) | \gamma(a) = A, \gamma(b) = B\}$ è l'universo di tutte le possibili curve di interesse.

Si noti in particolare che una curva $\gamma(t) \in \mathcal{U}$ è del tipo $\gamma(t) = (t, g(t))$ con $t \in [a, b]$

Definizione (Lunghezza di una curva di classe $C^1(a, b)$)

$$\mathcal{L}(\gamma) = \int_a^b \|\gamma'(t)\| dt \quad (2)$$

Definizione dell'universo di curve

$\mathcal{U} = \{\gamma \in C^1(a, b) | \gamma(a) = A, \gamma(b) = B\}$ è l'universo di tutte le possibili curve di interesse.

Dalla definizione di norma vettoriale, si ricava che la lunghezza di ogni curva $\gamma \in \mathcal{U}$ è data da:

$$\mathcal{L}(\gamma) = \int_a^b \sqrt{1 + g'(t)^2} dt \quad (3)$$

Definizione dell'universo di curve

$\mathcal{U} = \{\gamma \in C^1(a, b) | \gamma(a) = A, \gamma(b) = B\}$ è l'universo di tutte le possibili curve di interesse.

Dalla definizione di norma vettoriale, si ricava che la lunghezza di ogni curva $\gamma \in \mathcal{U}$ è data da:

$$\mathcal{L}(\gamma) = \int_a^b \sqrt{1 + g'(t)^2} dt \quad (3)$$

Obiettivo

Trovare la curva nell'insieme \mathcal{U} che minimizza il precedente valore $\mathcal{L}[\gamma]$;

Definizione (Funzionale)

Dato un universo di funzioni \mathcal{U} si definisce funzionale un'applicazione $J : \mathcal{U} \rightarrow \mathbb{R}$, che quindi associa ad una funzione un numero reale.

Definizione (Funzionale)

Dato un universo di funzioni \mathcal{U} si definisce funzionale un'applicazione $J : \mathcal{U} \rightarrow \mathbb{R}$, che quindi associa ad una funzione un numero reale.

Definizione (Minimante di un funzionale)

Sia $\hat{u} \in \mathcal{U}$ allora \hat{u} è minimante per il funzionale $J : \mathcal{U} \rightarrow \mathbb{R}$ se:

$$J[\hat{u}] \leq J[u] \quad \forall u \in \mathcal{U} \tag{4}$$

Nei casi da noi esaminati il funzionale J da minimizzare è sempre del tipo:

$$J[\phi] = \int_{\Omega \subset \mathbb{R}^n} f(x, \phi, \phi') dx \quad (5)$$

dove $\phi = \phi(x)$ e $\phi' = \phi'(x)$.

Nei casi da noi esaminati il funzionale J da minimizzare è sempre del tipo:

$$J[\phi] = \int_{\Omega \subset \mathbb{R}^n} f(x, \phi, \phi') dx \quad (5)$$

dove $\phi = \phi(x)$ e $\phi' = \phi'(x)$.

Problema

Trovare un parallelo tra le condizioni di minimo per funzioni reali e le condizioni di minimante di funzionali.

Definizione (Derivata Funzionale)

Dato un funzionale $J[\phi] = \int_{\Omega} f(x, \phi, \phi') dx$ si definisce derivata funzionale di J rispetto a ϕ il valore:

$$\frac{\delta J}{\delta \phi} = \frac{df}{d\phi} - \frac{d}{dx} \left(\frac{df}{d\phi'} \right);$$



Definizione (Derivata Funzionale)

Dato un funzionale $J[\phi] = \int_{\Omega} f(x, \phi, \phi') dx$ si definisce derivata funzionale di J rispetto a ϕ il valore:

$$\frac{\delta J}{\delta \phi} = \frac{df}{d\phi} - \frac{d}{dx} \left(\frac{df}{d\phi'} \right);$$

Un semplice esempio

Sia $J[\phi] = \int_a^b \phi(x)^2 + \phi'(x)x^4 dx$ allora

- $f(x, \phi, \phi') = \phi(x)^2 + \phi'(x)x^4$
- $\frac{\delta J}{\delta \phi} = 2\phi(x) - 4x^3;$

Equazione di Eulero-Lagrange

Proposizione

Sia $J : \mathcal{U} \rightarrow \mathbb{R}$; se $\phi \in \mathcal{U}$ è minimante per J allora:

$$\frac{\delta J}{\delta \phi} = \frac{df}{d\phi} - \frac{d}{dx} \left(\frac{df}{d\phi'} \right) = 0 \quad (6)$$

Equazione di Eulero-Lagrange

Proposizione

Sia $J : \mathcal{U} \rightarrow \mathbb{R}$; se $\phi \in \mathcal{U}$ è minimante per J allora:

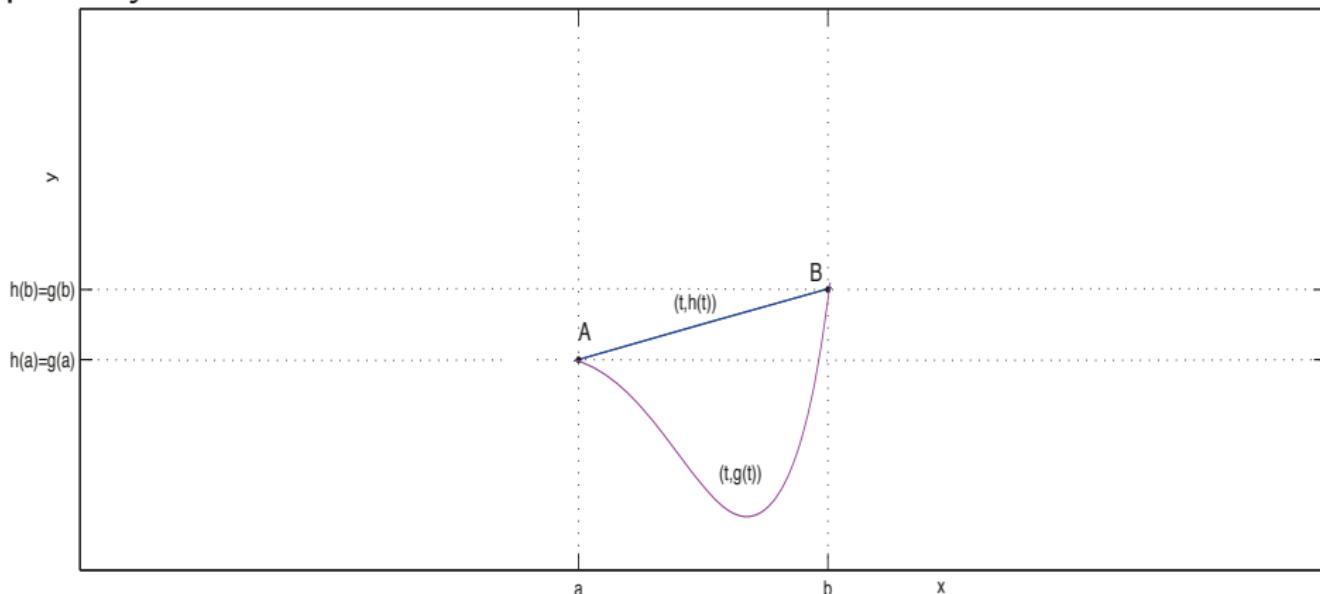
$$\frac{\delta J}{\delta \phi} = \frac{df}{d\phi} - \frac{d}{dx} \left(\frac{df}{d\phi'} \right) = 0 \quad (6)$$

Attenzione!

Questa una condizione necessaria ma non sufficiente. Per fornire una condizione sufficiente di minimante occorre verificare la convessità della funzione integranda f .

Esempio

Torniamo all'esempio della curva più corta che unisce i punti A e B sul piano xy .



Esempio continua... 1

Verifica

Si verifichi che il segmento, minimante per il funzionale:

$$\mathcal{L}[\gamma] = \int_a^b f(t, \gamma, \gamma') dt = \int_a^b \|\gamma'(t)\| dt \quad (7)$$

verifica l'equazione di Eulero-Lagrange.

Esempio continua... 1

Verifica

Si verifichi che il segmento, minimante per il funzionale:

$$\mathcal{L}[\gamma] = \int_a^b f(t, \gamma, \gamma') dt = \int_a^b \|\gamma'(t)\| dt \quad (7)$$

verifica l'equazione di Eulero-Lagrange.

La derivata della retta passante passante per A e B è il vettore $\gamma'(t) = (1, \frac{h(b)-h(a)}{b-a})$. sostituendola nell'equazione precedente si ha:

$$\mathcal{L}(\gamma) = \int_a^b \sqrt{1 + \left(\frac{h(b) - h(a)}{b - a} \right)^2} dt; \quad (8)$$

Esempio continua..2

Proposizione

Sia $J : \mathcal{U} \rightarrow \mathbb{R}$; se $\phi \in \mathcal{U}$ è minimante per J allora:

$$\frac{\delta J}{\delta \phi} = \frac{df}{d\phi} - \frac{d}{dx} \left(\frac{df}{d\phi'} \right) = 0$$

Poichè la funzione integranda, non dipende da γ , ma solo da γ' la derivata funzionale si riduce a:

$$\frac{\delta \mathcal{L}}{\delta \gamma} = - \frac{d}{dt} \frac{\delta f}{\delta \gamma'} \quad (9)$$

Proseguendo si ha:

$$\frac{\delta f}{\delta \gamma'} = \frac{1}{\|\gamma'(t)\|} \gamma'(t) \quad (10)$$

Ma $\frac{d}{dt} \frac{\delta f}{\delta \gamma'} = 0$ perchè $\gamma'(t)$ è costante rispetto a t e quindi si ha la tesi.

Tornando quindi alla formulazione di Kass, quella che vorremmo ottenere è una curva C minimante per il funzionale:

$$J[C] = \int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds$$

Tornando quindi alla formulazione di Kass, quella che vorremmo ottenere è una curva C minimante per il funzionale:

$$J[C] = \int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds$$

Problema di ricerca dell' ottimalità

Entrambi gli algoritmi che vedremo in seguito, tuttavia sfruttano solamente le condizioni necessarie di minimante di un funzionale non garantendo l'ottimo globale del problema.



Rappresentazione di curve

Obiettivo

Data un'immagine u_0 , l'obiettivo degli algoritmi basati sui *contorni attivi* o *snakes* è quello di far evolvere una curva soggetta a vincoli, allo scopo di determinare regioni e oggetti nell'immagine.

Rappresentazione curve nel piano

- esplicita / parametrica.



Rappresentazione di curve

Obiettivo

Data un'immagine u_0 , l'obiettivo degli algoritmi basati sui *contorni attivi* o *snakes* è quello di far evolvere una curva soggetta a vincoli, allo scopo di determinare regioni e oggetti nell'immagine.

Rappresentazione curve nel piano

- esplicita / parametrica.
- implicita / level set



Rappresentazione esplicita - Kass

Rappresentazione esplicita

Nella rappresentazione esplicita la curva che descrive il contorno dell'immagine viene espressa in forma parametrica $C : [0, 1] \rightarrow \mathbb{R}^2$.



Rappresentazione esplicita - Kass

Rappresentazione esplicita

Nella rappresentazione esplicita la curva che descrive il contorno dell'immagine viene espressa in forma parametrica $C : [0, 1] \rightarrow \mathbb{R}^2$.

Nella formulazione classica (M. Kass, A. Witkin, and D. Terzopoulos, 1988) il funzionale da minimizzare è :

$$\int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds \quad (11)$$



Rappresentazione esplicita - Kass

Rappresentazione esplicita

Nella rappresentazione esplicita la curva che descrive il contorno dell'immagine viene espressa in forma parametrica $C : [0, 1] \rightarrow \mathbb{R}^2$.

Nella formulazione classica (M. Kass, A. Witkin, and D. Terzopoulos, 1988) il funzionale da minimizzare è :

$$\int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds \quad (11)$$

con:

- $E_{\text{interna}} = \alpha \int_0^1 |C'(s)|^2 \, ds + \beta \int_0^1 |C''(s)| \, ds$



Rappresentazione esplicita - Kass

Rappresentazione esplicita

Nella rappresentazione esplicita la curva che descrive il contorno dell'immagine viene espressa in forma parametrica $C : [0, 1] \rightarrow \mathbb{R}^2$.

Nella formulazione classica (M. Kass, A. Witkin, and D. Terzopoulos, 1988) il funzionale da minimizzare è :

$$\int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds \quad (11)$$

con:

- $E_{\text{interna}} = \alpha \int_0^1 |C'(s)|^2 \, ds + \beta \int_0^1 |C''(s)| \, ds$
- $E_{\text{esterna}} = -\lambda \int_0^1 |\nabla u_0(C(s))|^2 \, ds$ dove u_0 è l'immagine



Rappresentazione esplicita - Kass

Rappresentazione esplicita

Nella rappresentazione esplicita la curva che descrive il contorno dell'immagine viene espressa in forma parametrica $C : [0, 1] \rightarrow \mathbb{R}^2$.

Nella formulazione classica (M. Kass, A. Witkin, and D. Terzopoulos, 1988) il funzionale da minimizzare è :

$$\int_0^1 E_{\text{snake}}(C(s)) \, ds = \int_0^1 E_{\text{interna}}(C(s)) \, ds + \int_0^1 E_{\text{esterna}}(C(s)) \, ds \quad (11)$$

con:

- $E_{\text{interna}} = \alpha \int_0^1 |C'(s)|^2 \, ds + \beta \int_0^1 |C''(s)| \, ds$
- $E_{\text{esterna}} = -\lambda \int_0^1 |\nabla u_0(C(s))|^2 \, ds$ dove u_0 è l'immagine
- $\alpha, \beta, \lambda \in \mathbb{R}$ dove $\alpha, \beta, \lambda \geq 0$

Rappresentazione esplicita - Kass

- $E_{interna}$ controlla la regolarità della curva.

Rappresentazione esplicita - Kass

- $E_{interna}$ controlla la regolarità della curva.
 - $E_{esterna}$ attira la curva a seguire il contorno dell'oggetto.

- $E_{interna}$ controlla la regolarità della curva.
 - $E_{esterna}$ attrae la curva a seguire il contorno dell'oggetto.

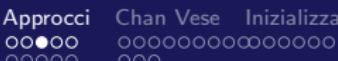
Minimizzare l'energia (11) è equivalente a posizionare la curva nei punti di massimo di $|\nabla u_0(C(s))|$ agendo come un *edge-detector*, mantenendo però la regolarità della curva.

Nelle implementazioni più usate la curva viene parametrizzata con una B-spline.

Rappresentazione esplicita - Kass

Gli svantaggi di questo approccio riguardano:

- impossibilità nell'ottenere curve con angoli e cuspidi. Nella definizione e nella minimizzazione del funzionale è richiesta una curva di classe C^2 .



Rappresentazione esplicita - Kass

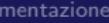
Gli svantaggi di questo approccio riguardano:

- impossibilità nell'ottenere curve con angoli e cuspidi. Nella definizione e nella minimizzazione del funzionale è richiesta una curva di classe C^2 .
- impossibilità di un cambio automatico della topologia della curva. Se gli oggetti da individuare identificano un insieme non连通的, la curva non sarà in grado di tracciarne il contorno.

Rappresentazione esplicita - Kass

Impossibilità di ottenere angoli e cuspidi

video : *StellaKass.mov*



Rappresentazione esplicita - Kass

Impossibilità di cambiare topologia della curva

video : DueCerchiKass.mov



Level Set Method

Level Set

Definizione

Sia $f : \Omega \rightarrow \mathbb{R}$, $\Omega \in \mathbb{R}^n$, si definisce insieme livello

$$L_c(f) = \{x \in \Omega | \phi(x) = c\}$$

Quando il numero di variabili è 2 l'insieme di livello è rappresentato da una curva, quando $n=3$ si parla di superficie di livello.

- Il gradiente della funzione è perpendicolare ad ogni punto della linea di livello.



Level Set Method

Level Set Method

- Level Set Method è stato sviluppato negli anni 80 da Stanley Osher and James Sethian [OS].



Level Set Method

Level Set Method

- Level Set Method è stato sviluppato negli anni 80 da Stanley Osher and James Sethian [OS].
- E' una tecnica numerica che consente di utilizzare curve e superfici senza aver bisogno di una parametrizzazione.



Level Set Method

Level Set Method

- Level Set Method è stato sviluppato negli anni 80 da Stanley Osher and James Sethian [OS].
- E' una tecnica numerica che consente di utilizzare curve e superfici senza aver bisogno di una parametrizzazione.
- In due dimensioni equivale a rappresentare una curva chiusa (nel nostro caso il contorno), utilizzando una funzione ausiliaria chiamata *level set function* ϕ .



Level Set Method

Level Set Method

- Level Set Method è stato sviluppato negli anni 80 da Stanley Osher and James Sethian [OS].
- E' una tecnica numerica che consente di utilizzare curve e superfici senza aver bisogno di una parametrizzazione.
- In due dimensioni equivale a rappresentare una curva chiusa (nel nostro caso il contorno), utilizzando una funzione ausiliaria chiamata *level set function* ϕ .
- Linea di livello 0: $C = \{(x, y) | \phi(x, y) = 0\}$



Level Set Method

Come evolve la linea di livello 0 di una funzione $\phi(x) + c$ al variare di c . video:
[levelset.mov](#)



Level Set Method

Proprietà

- Consente di ottenere curve con irregolarità quali angoli e cuspidi.



Proprietà

- Consente di ottenere curve con irregolarità quali angoli e cuspidi.
- Consente di seguire il cambiamento topologico dell'insieme di livello nel tempo.



Proprietà

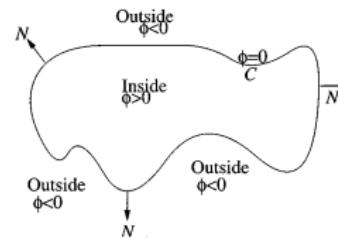
- Consente di ottenere curve con irregolarità quali angoli e cuspidi.
- Consente di seguire il cambiamento topologico dell'insieme di livello nel tempo.
- Manipola implicitamente C attraverso la funzione ϕ .

Proprietà

- Consente di ottenere curve con irregolarità quali angoli e cuspidi.
- Consente di seguire il cambiamento topologico dell'insieme di livello nel tempo.
- Manipola implicitamente C attraverso la funzione ϕ .
- Per convenzione $\phi > 0$ all'interno della regione delimitata da C ,
 $\phi < 0$ all'esterno.

Proprietà

- Consente di ottenere curve con irregolarità quali angoli e cuspidi.
- Consente di seguire il cambiamento topologico dell'insieme di livello nel tempo.
- Manipola implicitamente C attraverso la funzione ϕ .
- Per convenzione $\phi > 0$ all'interno della regione delimitata da C , $\phi < 0$ all'esterno.
- L'evoluzione della curva è data dalla linea di livello 0 al tempo t della funzione $\phi(t, x, y)$.





Level Set Method

Chan-vese cuspidi e cambiamento di topologia

Questi risultati sono stati ottenuti applicando l'algoritmo di segmentazione Chan-Vese che sfrutta il *level set method*.

Figure : Stella segmentata

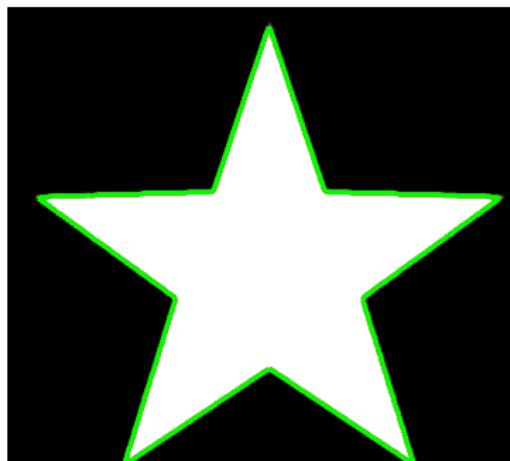
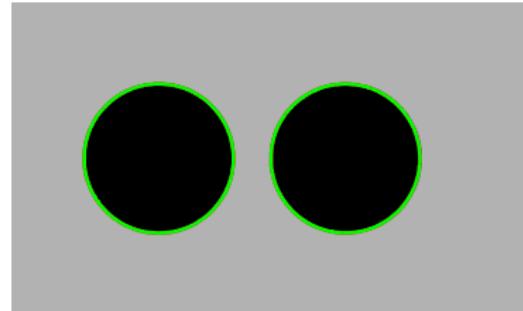


Figure : I due cerchi segmentati





Definizione del modello

ChanVase [CV2001]

Definiamo :

- $\omega \subset \Omega$ aperto.



Definizione del modello

ChanVase [CV2001]

Definiamo :

- $\omega \subset \Omega$ aperto.
- La curva $C = \partial\omega$



Definizione del modello

ChanVase [CV2001]

Definiamo :

- $\omega \subset \Omega$ aperto.
- La curva $C = \partial\omega$
- $inside(C) = \omega$ e $outside(C) = \Omega - \bar{\omega}$



Definizione del modello

ChanVase [CV2001]

Definiamo :

- $\omega \subset \Omega$ aperto.
- La curva $C = \partial\omega$
- $inside(C) = \omega$ e $outside(C) = \Omega - \bar{\omega}$
- $u_0(x, y)$ l'intensità dell'immagine nel pixel (x, y)



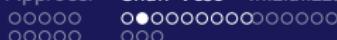
Definizione del modello

ChanVase [CV2001]

Definiamo :

- $\omega \subset \Omega$ aperto.
- La curva $C = \partial\omega$
- $inside(C) = \omega$ e $outside(C) = \Omega - \bar{\omega}$
- $u_0(x, y)$ l'intensità dell'immagine nel pixel (x, y)

La segmentazione avviene tramite la minimizzazione di un funzionale di energia.



Definizione del modello

$$F_1(C) + F_2(C) = \int_{inside(C)} |u_0(x, y) - c_1|^2 dx dy + \int_{outside(C)} |u_0(x, y) - c_2|^2 dx dy$$

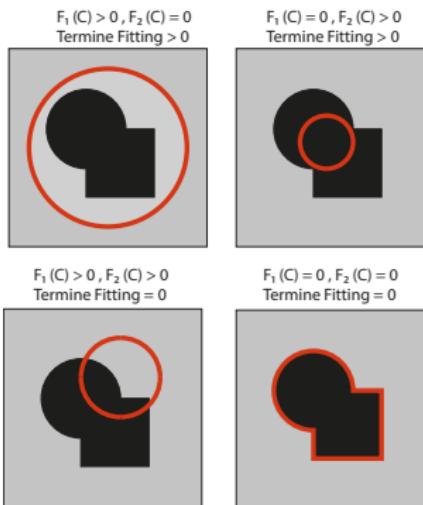
- c_1 è la media dell'immagine u_0 dentro C .



Definizione del modello

$$F_1(C) + F_2(C) = \int_{inside(C)} |u_0(x, y) - c_1|^2 dx dy + \int_{outside(C)} |u_0(x, y) - c_2|^2 dx dy$$

- c_1 è la media dell'immagine u_0 dentro C .
- c_2 è la media dell'immagine u_0 fuori C .



Il valore minimo di $F_1(C) + F_2(C)$ si ottiene solo quando C è sul contorno





Definizione del modello

- Oltre al precedente termine di fitting, aggiungiamo dei termini di regolarizzazione, quali la lunghezza di C e l'area interna delimitata da C .

Definizione del modello

- Oltre al precedente termine di fitting, aggiungiamo dei termini di regolarizzazione, quali la lunghezza di C e l'area interna delimitata da C .

Otteniamo quindi:

$$\begin{aligned}
 F(c_1, c_2, C) = & \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(\text{inside}(C)) + \\
 & + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy \\
 & + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy
 \end{aligned} \tag{12}$$

ricordando la notazione con il level set method:

$$\left\{ \begin{array}{l} C = \partial\omega = \{(x, y) \in \omega : \phi(x, y) = 0\} . \end{array} \right.$$

Definizione del modello

- Oltre al precedente termine di fitting, aggiungiamo dei termini di regolarizzazione, quali la lunghezza di C e l'area interna delimitata da C .

Otteniamo quindi:

$$\begin{aligned}
 F(c_1, c_2, C) = & \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(\text{inside}(C)) + \\
 & + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy \\
 & + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy
 \end{aligned} \tag{12}$$

ricordando la notazione con il level set method:

$$\begin{cases} C = \partial\omega = \{(x, y) \in \omega : \phi(x, y) = 0\} . \\ \text{inside}(C) = \omega = \{(x, y) \in \omega : \phi(x, y) > 0\} . \end{cases}$$

Definizione del modello

- Oltre al precedente termine di fitting, aggiungiamo dei termini di regolarizzazione, quali la lunghezza di C e l'area interna delimitata da C .

Otteniamo quindi:

$$\begin{aligned}
 F(c_1, c_2, C) = & \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(\text{inside}(C)) + \\
 & + \lambda_1 \int_{\text{inside}(C)} |u_0(x, y) - c_1|^2 dx dy \\
 & + \lambda_2 \int_{\text{outside}(C)} |u_0(x, y) - c_2|^2 dx dy
 \end{aligned} \tag{12}$$

ricordando la notazione con il level set method:

$$\begin{cases} C = \partial\omega = \{(x, y) \in \omega : \phi(x, y) = 0\} . \\ \text{inside}(C) = \omega = \{(x, y) \in \omega : \phi(x, y) > 0\} . \\ \text{outside}(C) = \omega = \{(x, y) \in \omega : \phi(x, y) < 0\} . \end{cases}$$

Definizione del modello

Introducendo le funzioni di Heaviside e la delta di Dirac, per ottenere lo stesso segno di integrale

$$H(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

$$\delta(z) = \frac{d}{dz} H(z)$$
 otteniamo:

- $\text{Length } \{\phi = 0\} = \int_{\Omega} |\nabla H(\phi(x, y))| dx dy = \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy$

Definizione del modello

Introducendo le funzioni di Heaviside e la delta di Dirac, per ottenere lo stesso segno di integrale

$$H(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

$$\delta(z) = \frac{d}{dz} H(z)$$
 otteniamo:

- $\text{Length } \{\phi = 0\} = \int_{\Omega} |\nabla H(\phi(x, y))| dx dy = \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy$
- $\text{Area } \{\phi \geq 0\} = \int_{\Omega} H(\phi(x, y)) dx dy$

Definizione del modello

Introducendo le funzioni di Heaviside e la delta di Dirac, per ottenere lo stesso segno di integrale

$$H(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

$\delta(z) = \frac{d}{dz} H(z)$ otteniamo:

- $Length \{\phi = 0\} = \int_{\Omega} |\nabla H(\phi(x, y))| dx dy = \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy$
- $Area \{\phi \geq 0\} = \int_{\Omega} H(\phi(x, y)) dx dy$
- $\int_{\phi > 0} |u_0(x, y) - c_1|^2 dx dy = \int_{\Omega} |u_0(x, y) - c_1|^2 H(\phi(x, y)) dx dy$



Definizione del modello

Introducendo le funzioni di Heaviside e la delta di Dirac, per ottenere lo stesso segno di integrale

$$H(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

$$\delta(z) = \frac{d}{dz} H(z)$$

otteniamo:

- $\text{Length } \{\phi = 0\} = \int_{\Omega} |\nabla H(\phi(x, y))| dx dy = \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy$
- $\text{Area } \{\phi \geq 0\} = \int_{\Omega} H(\phi(x, y)) dx dy$
- $\int_{\phi > 0} |u_0(x, y) - c_1|^2 dx dy = \int_{\Omega} |u_0(x, y) - c_1|^2 H(\phi(x, y)) dx dy$
- $\int_{\phi < 0} |u_0(x, y) - c_2|^2 dx dy = \int_{\Omega} |u_0(x, y) - c_2|^2 (1 - H(\phi(x, y))) dx dy$



Definizione del modello

Quindi il funzionale diventa:

$$\begin{aligned}
 F(c_1, c_2, \phi) = & \mu \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy + \\
 & + \nu \int_{\Omega} H(\phi(x, y)) dx dy + \\
 & + \lambda_1 \int_{\Omega} |u_0(x, y) - c_1|^2 H(\phi(x, y)) dx dy + \\
 & + \lambda_2 \int_{\Omega} |u_0(x, y) - c_2|^2 (1 - H(\phi(x, y))) dx dy
 \end{aligned} \tag{13}$$

- Osservare che in questa definizione non viene usato il gradiente dell'immagine. Contorni attivi senza edge.



Definizione del modello

Fissando ϕ e minimizzando rispetto a c_1 otteniamo :

$$\int_{\Omega} 2u_0(x, y)H(\phi(x, y)) \, dx \, dy - \int_{\Omega} 2c_1 H(\phi(x, y)) \, dx \, dy = 0 \quad (14)$$

quindi:

- $c1(\phi) = \frac{\int_{\Omega} u_0(x, y)H(\phi(x, y)) \, dx \, dy}{\int_{\Omega} H(\phi(x, y)) \, dx \, dy}$ media di u_0 all'interno della curva.



Definizione del modello

Fissando ϕ e minimizzando rispetto a c_1 otteniamo :

$$\int_{\Omega} 2u_0(x, y)H(\phi(x, y)) \, dx \, dy - \int_{\Omega} 2c_1 H(\phi(x, y)) \, dx \, dy = 0 \quad (14)$$

quindi:

- $c1(\phi) = \frac{\int_{\Omega} u_0(x, y)H(\phi(x, y)) \, dx \, dy}{\int_{\Omega} H(\phi(x, y)) \, dx \, dy}$ media di u_0 all'interno della curva.
- $c2(\phi) = \frac{\int_{\Omega} u_0(x, y)(1-H(\phi(x, y))) \, dx \, dy}{\int_{\Omega} (1-H(\phi(x, y))) \, dx \, dy}$ media di u_0 all'esterno della curva.



Definizione del modello

Dato il funzionale:

$$\begin{aligned}
 F(c_1, c_2, \phi) = & \mu \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy + \\
 & + \nu \int_{\Omega} H(\phi(x, y)) dx dy + \\
 & + \lambda_1 \int_{\Omega} |u_0(x, y) - c_1|^2 H(\phi(x, y)) dx dy + \\
 & + \lambda_2 \int_{\Omega} |u_0(x, y) - c_1|^2 (1 - H(\phi(x, y))) dx dy
 \end{aligned} \tag{15}$$

si può dimostrare che:

$$\frac{\delta F}{\delta \phi} = \delta(\phi) \left[-\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \nu + \lambda_1 (u_0 - c_1)^2 - \lambda_2 (u_0 - c_2)^2 \right] \tag{16}$$

Definizione del modello

Ricordando che un minimante per un funzionale deve soddisfare l'equazione di Eulero-Lagrange, allora fissando c_1 e c_2 e minimizzando rispetto a ϕ :

$$\delta(\phi) \left[-\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + \nu + \lambda_1(u_0 - c_1)^2 - \lambda_2(u_0 - c_2)^2 \right] = 0 \quad (17)$$

in $(0, \infty) \times \Omega$

Come far evolvere la funzione ϕ nel tempo?

Una possibile soluzione è quella di usare il metodo del gradiente, ricordando che l'antigradiente fornisce sempre una buona direzione di discesa.



Definizione del modello

Regola di aggiornamento della funzione ϕ

$$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2 \right] \quad (18)$$

Problema per implementazione

Finora abbiamo lavorato come se lo spazio sul quale è definita l'immagine sia continuo; dobbiamo traslare questo metodo su un insieme discreto di valori.



Implementazione

Differenze finite

- Conoscendo ϕ^n inizialmente sono calcolate $c_1(\phi)$ e $c_2(\phi)$, media interna e esterna.
- Successivamente viene approssimata la curvatura mediante le differenze finite

$$\operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) = \frac{\phi_{xx}\phi_y^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_x^2}{(\phi_x^2 + \phi_y^2)^{\frac{3}{2}}} \text{ dove :}$$

$$\begin{cases} \phi_x = \frac{\phi_{x+\Delta x,y} - \phi_{x,y}}{\Delta x} \\ \phi_y = \frac{\phi_{x,y+\Delta y} - \phi_{x,y}}{\Delta y} \\ \phi_{x,y} = \frac{\phi_{x+\Delta x,y+\Delta y} - \phi_{x,y+\Delta y} - (\phi_{x+\Delta x,y} - \phi_{x,y})}{\Delta x \Delta y} \end{cases}$$

con $\Delta x = \Delta y = 1$;

Implementazione

Regola di aggiornamento

$$\begin{aligned}\phi_{i,j}^{n+1} = & \phi_{i,j}^n + \\& \delta(\phi) \left(\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) (i,j) - \nu - \lambda_1(u_0(i,j) - c_1)^2 - \lambda_2(u_0(i,j) - c_2)^2 \right) \Delta t\end{aligned}$$



Implementazione

Dati $n = 0; \phi^n = \phi_0$

while soluzione ϕ^n non è stazionaria **do**

calcola $c_1(\phi^n)$ e $c_2(\phi^n)$

calcola ϕ^{n+1}

n = n +1;

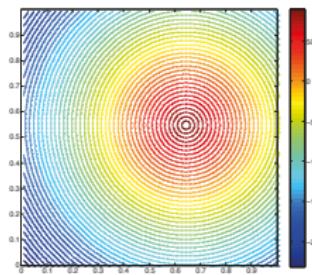
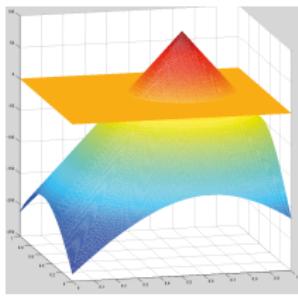
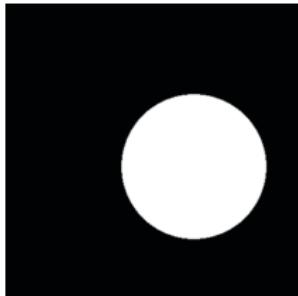
end

Inizializzazione

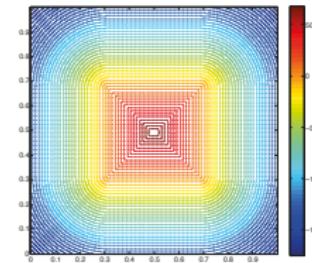
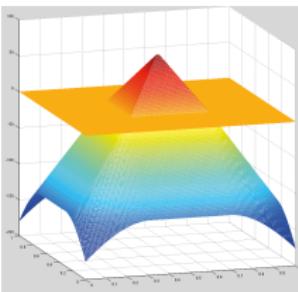
- Il punto dell'algoritmo del quale non abbiamo ancora parlato è il passo dell'inizializzazione.
- Ricordando la formulazione con il *level set method* non è necessaria una parametrizzazione del contorno, ma una funzione di supporto ϕ .
- Il luogo degli zeri di ϕ costituisce il contorno iniziale.
- Un'immagine è un insieme discreto di punti.
- Data una maschera iniziale, il cui perimetro rappresenta il contorno iniziale, viene inizializzata ϕ^0 come distanza segnata dal contorno stesso.
- Le maschere più usate sono quelle sferiche e rettangolari.



Maschera Iniziale: Sferica



Maschera Iniziale: Quadrata



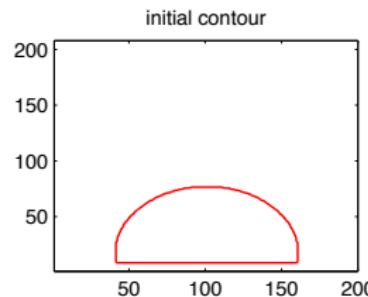
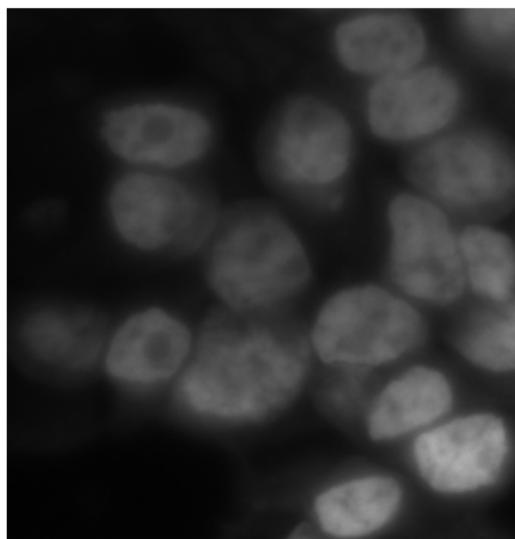
oooooo
oooooo

oo
ooo
ooo
o

- E' bene ricordare che l'algoritmo usa la tecnica di discesa del gradiente.
- L'algoritmo trova soluzioni locali e non globali.
- Cambiare punto di partenza (ϕ^0) può far ottenere una segmentazione differente dell'immagine.

Alcuni Esempi su immagini biomediche

Alcuni Esempi

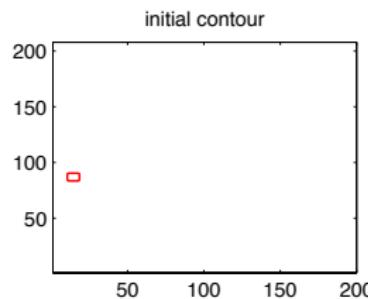
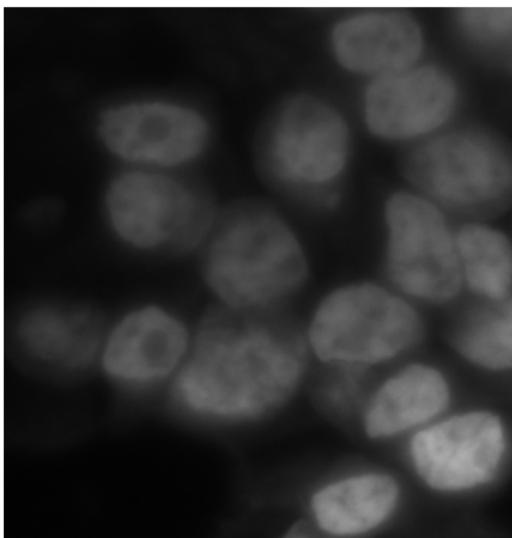


Global Region-Based Segmentation



Alcuni Esempi su immagini biomediche

Alcuni Esempi

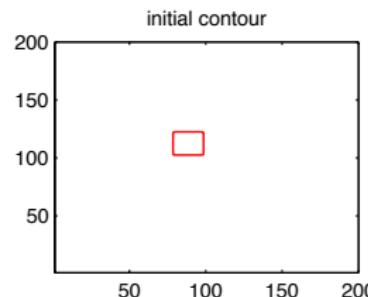


Global Region-Based Segmentation



Alcuni Esempi su immagini biomediche

Alcuni Esempi

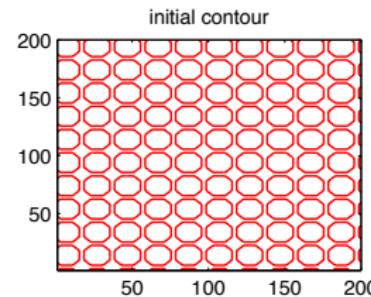


Global Region-Based Segmentation



Alcuni Esempi su immagini biomediche

Alcuni Esempi

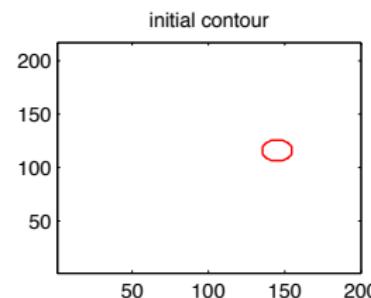
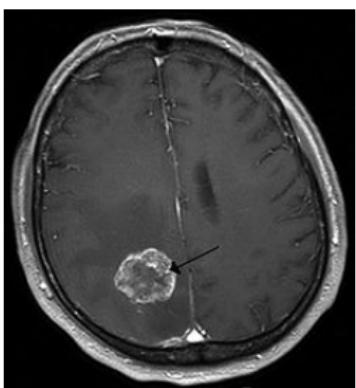


Global Region-Based Segmentation

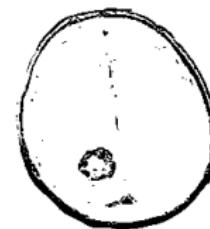


Alcuni Esempi su immagini biomediche

Alcuni Esempi

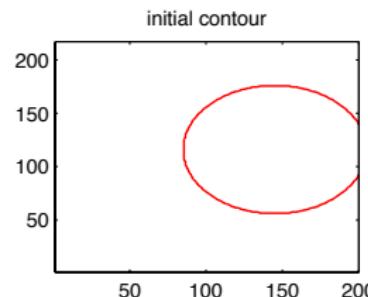
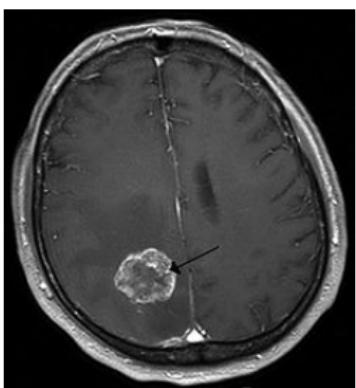


Global Region-Based Segmentation



Alcuni Esempi su immagini biomediche

Alcuni Esempi



Global Region-Based Segmentation



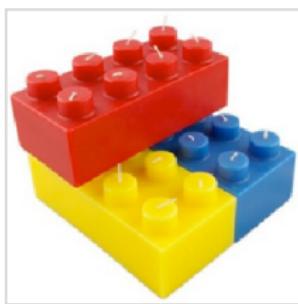
Chan Vese Vector Image [CV2002]

- Abbiamo parlato di intensità di colore, considerando l'immagine sempre in scala di grigi.
- Cosa può accadere per immagini a colore?



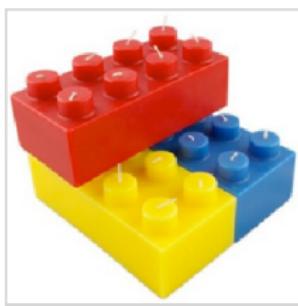
Chan Vese Vector Image [CV2002]

- Abbiamo parlato di intensità di colore, considerando l'immagine sempre in scala di grigi.
- Cosa può accadere per immagini a colore?



Chan Vese Vector Image [CV2002]

- Abbiamo parlato di intensità di colore, considerando l'immagine sempre in scala di grigi.
- Cosa può accadere per immagini a colore?



- Potrei pensare di applicare l'algoritmo a ogni canale dell'immagine.
- Otterrei diversi profili, uno per ogni canale.



Come diventa il funzionale da minimizzare?

$$\begin{aligned}
 F(c^+, c^-, \phi) = & \mu \int_{\Omega} \delta(\phi(x, y)) |\nabla \phi(x, y)| dx dy + \nu \int_{\Omega} H(\phi(x, y)) dx dy \\
 & + \frac{1}{N} \int_{\Omega} \sum_{i=1}^N \lambda_i^+ |u_{0,i}(x, y) - c_i^+|^2 H(\phi(x, y)) dx dy
 \end{aligned} \tag{19}$$

$$+ \frac{1}{N} \int_{\Omega} \sum_{i=1}^N \lambda_i^- |u_{0,i}(x, y) - c_i^-|^2 (1 - H(\phi(x, y))) dx dy$$

i = 1...N numero di canali

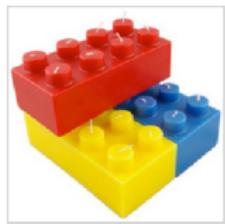
$$c_i^+ = \frac{\int_{\Omega} u_{0,i}(x, y) H(\phi(x, y)) dx dy}{\int_{\Omega} H(\phi(x, y)) dx dy}$$

$$c_i^- = \frac{\int_{\Omega} u_{0,i}(x, y) (1 - H(\phi(x, y))) dx dy}{\int_{\Omega} (1 - H(\phi(x, y))) dx dy}$$

oooooo

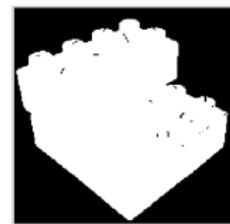
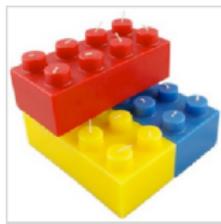
oooooooooooooooooooo

oo
ooo
ooo
o



oooooo

oooooooooooooooooooo

oo
ooo
ooo
o

oooooo

oooooo

oooooooooooooooooooo

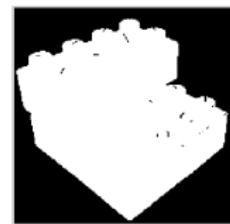
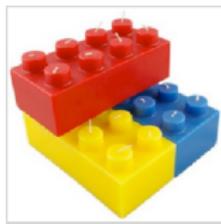
Vector

Multiphase

Sperimentazione

Bibliografia

Lab.

oo
ooo
ooo
o

- Funzione anche per immagini rumorose, a meno che il rumore non abbia contaminato tutti i vari canali dell'immagine.

oooooo

oooooo

oooooooooooooooooooo

Vector

Multiphase

Sperimentazione

Bibliografia

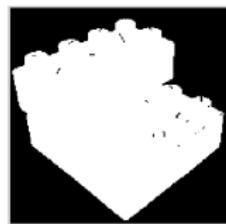
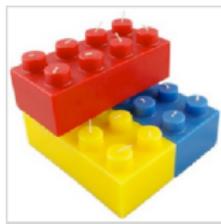
Lab.

oo

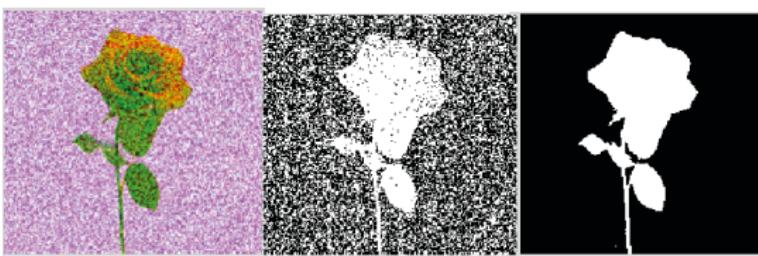
ooo

ooo

o



- Funzione anche per immagini rumorose, a meno che il rumore non abbia contaminato tutti i vari canali dell'immagine.



oooooo

oooooo

oooooooooooooooooooo

Vector

Multiphase

Sperimentazione

Bibliografia

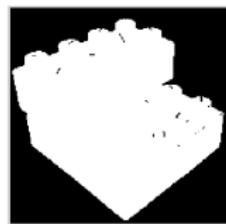
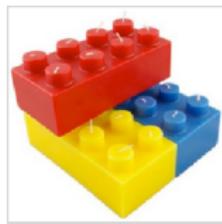
Lab.

oo

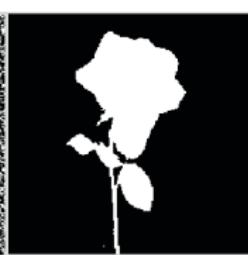
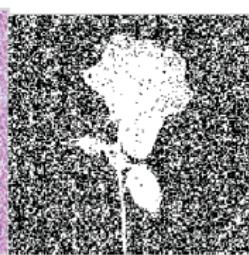
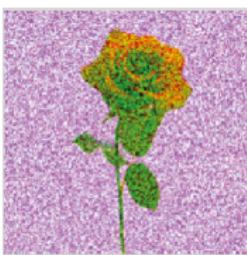
ooo

ooo

o



- Funzione anche per immagini rumorose, a meno che il rumore non abbia contaminato tutti i vari canali dell'immagine.



- Se ho immagini prive di rumore, posso utilizzare l'algoritmo che utilizza un solo canale.

oooooo
oooooo

oo
ooo
ooo
o

Chan Vese Multiphase

- Con i precedenti approcci possiamo distinguere il background dal foreground.

oooooo
oooooo

oo
ooo
ooo
o

Chan Vese Multiphase

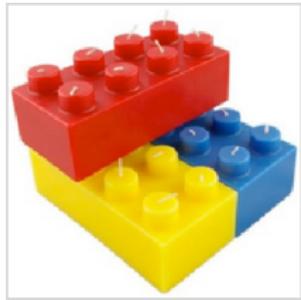
- Con i precedenti approcci possiamo distinguere il background dal foreground.
- Per qualsiasi immagine con più di due colori non riuscirò a catturare le varie differenze di tonalità.

oooooo
oooooo
ooooo

oo
ooo
ooo
o

Chan Vese Multiphase

- Con i precedenti approcci possiamo distinguere il background dal foreground.
- Per qualsiasi immagine con più di due colori non riuscirò a catturare le varie differenze di tonalità.

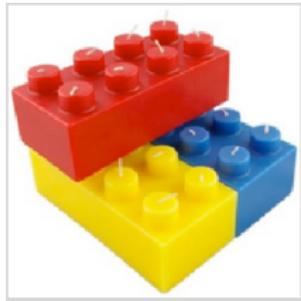


oooooo ooooooooooooooooooooo

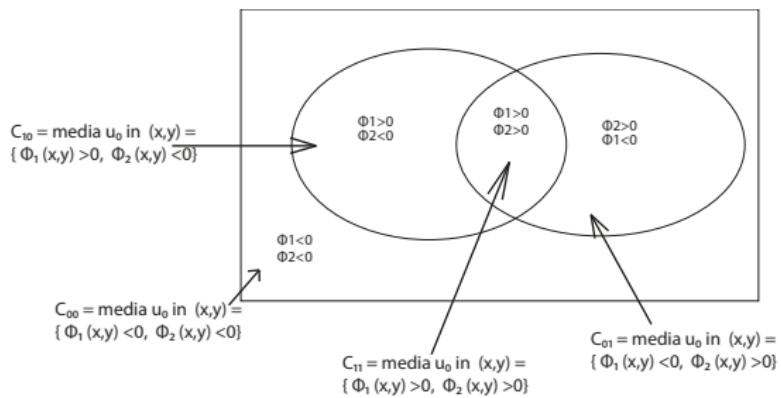
oo
ooo
ooo
o

Chan Vese Multiphase

- Con i precedenti approcci possiamo distinguere il background dal foreground.
- Per qualsiasi immagine con più di due colori non riuscirò a catturare le varie differenze di tonalità.



- Se ho N contorni posso distinguere 2^N regioni differenti.
 - Aumentare il numero della fasi N (numero Level Set Function).



$$\begin{aligned} F(c, \phi) = & \mu \int_{\Omega} |\nabla H(\phi_1)| dx dy + \mu \int_{\Omega} |\nabla H(\phi_2)| dx dy \\ & + \nu \int_{\Omega} H(\phi_1) dx dy + \nu \int_{\Omega} H(\phi_2) dx dy \\ & + \int_{\Omega} |u_0 - c_{10}|^2 H(\phi_1)(1 - H(\phi_2)) dx dy \\ & + \int_{\Omega} |u_0 - c_{11}|^2 H(\phi_1)H(\phi_2) dx dy \\ & + \int_{\Omega} |u_0 - c_{01}|^2 (1 - H(\phi_1))H(\phi_2) dx dy \\ & + \int_{\Omega} |u_0 - c_{00}|^2 (1 - H(\phi_1))(1 - H(\phi_2)) dx dy \end{aligned} \tag{20}$$

oooooo

oooooo

oooooooooooooooooooo

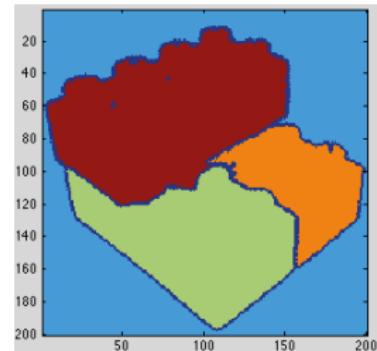
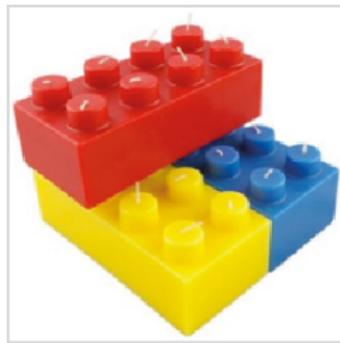
Vector

Multiphase

Sperimentazione

Bibliografia

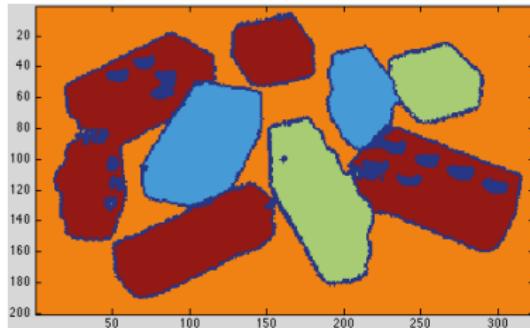
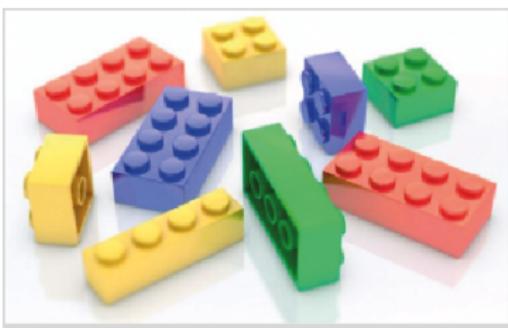
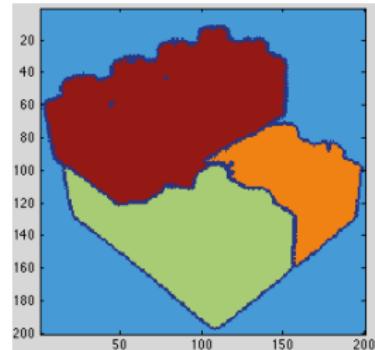
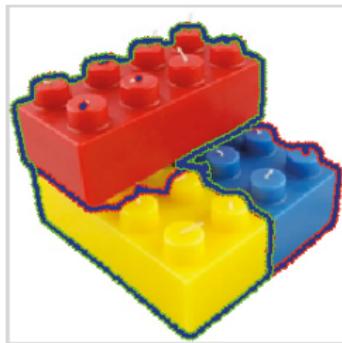
Lab.

oo
ooo
ooo
o

oooooo
oooooo

oooooooooooooooo

oo
ooo
ooo
o



Sperimentazione

Ricordando la formulazione del funzionale:

$$\begin{aligned} F(c, \phi) = & \mu \int_{\Omega} |\nabla H(\phi)| dx dy + \nu \int_{\Omega} H(\phi) dx dy \\ & + \lambda_1 \int_{\Omega} |u_0 - c_1|^2 H(\phi) dx dy + \lambda_2 \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi)) dx dy \end{aligned}$$

- Per il corretto funzionamento dell'algoritmo è necessaria la regolazione di quattro parametri $\lambda_1, \lambda_2, \mu, \nu$.
- Nella letteratura e nella maggior parte delle implementazioni esistenti un'impostazione comune dei parametri è la seguente:
 $\lambda_1 = \lambda_2 = 1, \mu = 0.2, \nu = 0$.
- Tale scelta è dovuta essenzialmente sulla base dei risultati ottenuti.

Parametro: λ_1

Parametro λ_1

Ricordando la formulazione del funzionale:

$$\begin{aligned}
F(c, \phi) = & \mu \int_{\Omega} |\nabla H(\phi)| dx dy + \nu \int_{\Omega} H(\phi) dx dy \\
& + \lambda_1 \int_{\Omega} |u_0 - c_1|^2 H(\phi) dx dy + \lambda_2 \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi)) dx dy
\end{aligned}$$

Considerazioni su λ_1

- λ_1 pesa l'addendo del funzionale che rappresenta la differenza al quadrato tra la media interna al contorno (intensità) e i pixel interni.
- Aumentando λ_1 l'algoritmo tenderà a segmentare regioni il più possibile uniformi all'interno (valgono di più), accettando una variazione maggiore all'esterno.

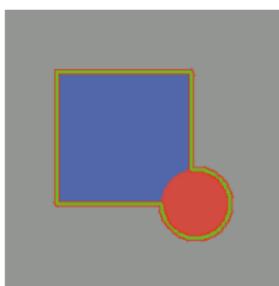
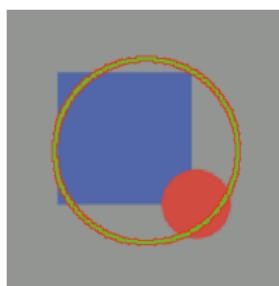
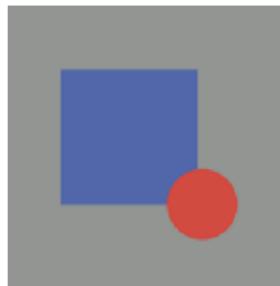
oooooo
oooooo
ooo

○●
○○○
○○○
○

Parametro: λ_1

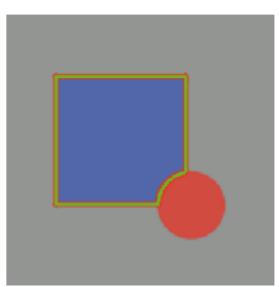
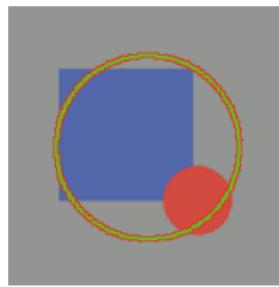
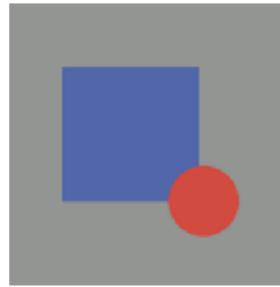
Esecuzione con parametri "standard" $\lambda_1 = \lambda_2 = 1$ $\mu = 0.2$ $v = 0$

Maschera iniziale: media.



Esecuzione con parametri $\lambda_1 = 30$ $\lambda_2 = 1$ $\mu = 0.2$ $v = 0$

Maschera iniziale: media.



Parametro: λ_2

Parametro λ_2

Ricordando la formulazione del funzionale:

$$\begin{aligned}
F(c, \phi) = & \mu \int_{\Omega} |\nabla H(\phi)| dx dy + \nu \int_{\Omega} H(\phi) dx dy \\
& + \lambda_1 \int_{\Omega} |u_0 - c_1|^2 H(\phi) dx dy + \lambda_2 \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi)) dx dy
\end{aligned}$$

Considerazioni su λ_2

- λ_2 pesa l'addendo del funzionale che rappresenta la differenza al quadrato tra la media esterna al contorno (intensità) e i pixel esterni.
- Aumentando λ_2 l'algoritmo tenderà a segmentare regioni il più possibile uniformi all'esterno (valgono di più), accettando una variazione maggiore all'interno.

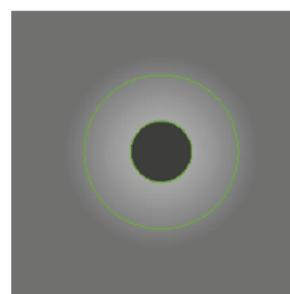
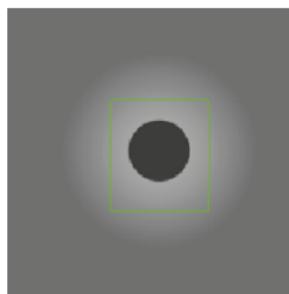
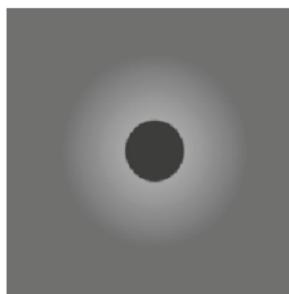
oooooo
oooooo

oooooooooooooooo

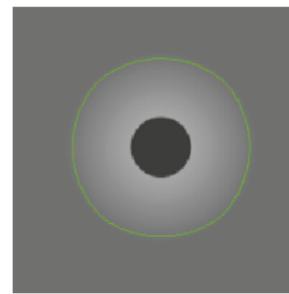
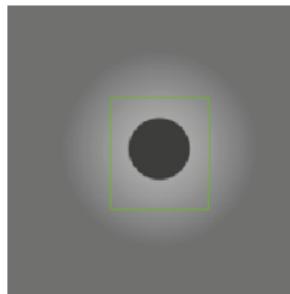
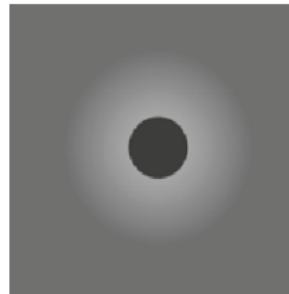
○○
○●○
○○○
○

Parametro: λ_2

Esecuzione con parametri "standard" $\lambda_1 = 1 \lambda_2 = 1 \mu = 0.2 v=0$



Esecuzione con parametri $\lambda_1 = 1 \lambda_2 = 100 \mu = 0.2 v=0$



oooooo
oooooo
ooooo

oo
oo●
oo○
o

Parametro: λ_2

Quindi:

- λ_1 e λ_2 sono due parametri di regolazione rispettivamente della forza interna ed esterna.
- Solo il rapporto tra i due gioca un ruolo centrale.

Parametro μ Parametro μ

Ricordiamo nuovamente la formulazione del funzionale:

$$\begin{aligned} F(c, \phi) = & \mu \int_{\Omega} |\nabla H(\phi)| dx dy + \nu \int_{\Omega} H(\phi) dx dy \\ & + \lambda_1 \int_{\Omega} |u_0 - c_1|^2 H(\phi) dx dy + \lambda_2 \int_{\Omega} |u_0 - c_2|^2 (1 - H(\phi)) dx dy \end{aligned}$$

Considerazioni su μ

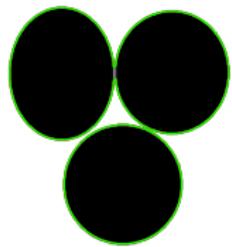
- μ controlla la lunghezza del contorno.
- Attribuendo un valore elevato a μ , la curva tende a minimizzare maggiormente la sua lunghezza piuttosto che la media interna ed esterna.
- Se μ è grande la curva tenderà a contornare gruppi di oggetti simili piuttosto che contornare singoli oggetti.

oooooo
oooooo
ooooo

○○
○○○
○●○
○

Parametro μ Parametro μ

500 Iterations



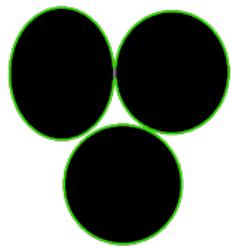
oooooo ooooooooooooooooooooo

○○
○○○
○●○
○

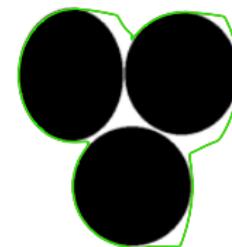
Parametro μ

Parametro μ

500 Iterations



500 Iterations

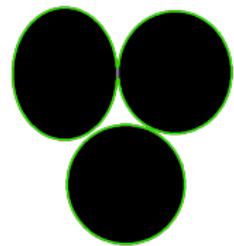


oooooo ooooooooooooooooooooo

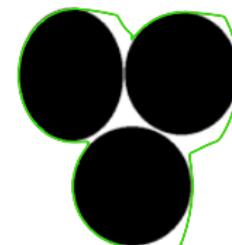
○○
○○○
○●○
○

Parametro μ Parametro μ

500 Iterations



500 Iterations



oooooo ooooooooooooooooooooo

○○
○○○
○○●
○

Parametro μ

- Variando il parametro della lunghezza spesso non si ottengono modiche salienti nella segmentazione finale.
- μ fornisce all'algoritmo la capacità anti-rumore.
- Evita di segmentare regioni troppo piccole e di ottenere contorni eccessivamente “ingarbugliati”.

Parametro ν Parametro ν

- Teoricamente aumentare il parametro ν porterebbe ad ottenere regioni segmentate con area inferiore.
- Tuttavia il comportamento dell'algoritmo è risultato instabile alla modifica del parametro ν .
- Per piccole variazioni il risultato è esattamente identico al caso $\nu = 0$.
- Per variazioni più consistenti il contorno degenera in un punto.
- Nella formalizzazione matematica compare anche il termine dell'area.
- Nella pratica non viene usato.

Conclusioni sui parametri

Conclusioni sui parametri

- La regolazione dei parametri non è facile.
- Se il risultato ottenuto non fosse soddisfacente è necessario regolare manualmente i parametri.
- Tuttavia non vi è alcuna garanzia di ottenere sempre dei risultati migliori, anzi molto spesso peggiorano.
- Una valida alternativa è quella di rieseguire l'algoritmo variando la maschera (contorno) iniziale.

Bibliografia

-  Chan, T.F & Vese, L.A. (2001). Active Contours Without Edges. *IEEE Transactions, Image Processing*. vol. 10 (2) pp.267-277.
-  Chan, T.F & Sandberg, L.A (2000). Active Contours Without Edges for Vector-Valued Images. *Journal of Visual Communication and Image Representation* vol 11 (2), pp. 130-141.
-  Chan, T.F & Vese (2002). A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *International Journal of Computer Vision*. Vol 50 (3) pp 271-293.
-  Kass, M. & Witkin, T. D. (1988). Snakes: Active Contour Models. *International Journal of Computer Vision* vol 1 (4) pp 321-331.
-  S. Osher & J. Sethian (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi



oooooo

oooooooooooooooooooo

oo
ooo
ooo
o

LABORATORIO

Algoritmo di Kass

Il prototipo della funzione è **function [P,J] = Snake2D (I,P,Options)**

Esempio sulla derivabilità del contorno

```
I = imread('immaginiTest/stella.png');  
I = im2double(I);  
figure, imshow(I); [y,x] = getpts;  
P=[x(:) y(:)];  
Options=struct;  
Options.Verbose=true;  
Options.Iterations=700;  
Options.Wterm = 0;  
Options.Wline = 0;  
O,J=Snake2D(I,P,Options);
```

oooooo ooooooooooooooooooooo

oo
ooo
ooo
o

Algoritmo di Kass

Provare a usare un contorno iniziale che avvolga entrambi i cerchi.

```
I = imread('immaginiTest/dueCerchi.png');
```

```
I = im2double(I);
```

```
figure, imshow(I); [y,x] = getpts;
```

```
P=[x(:) y(:)];
```

```
O,J=Snake2D(I,P,Options);
```

Adesso usare un contorno che avvolga solo un cerchio

```
figure, imshow(I); [y,x] = getpts;
```

```
P=[x(:) y(:)];
```

```
O,J=Snake2D(I,P,Options);
```

Impossibile cambiare topologia di curva

Algoritmo di Chan Vese

Il prototipo della funzione è **function seg = chenvese(I, mask, numIter, mu, lambda1, lambda2, vi, reverseColor, method)**

- I = immagine.
- $mask$ = maschera che determina il contorno iniziale. Può essere definita dall'utente, oppure è possibile sceglierne una predefinita: *small, medium, large, whole*.
- $numIter$ = numero di iterazioni dell'algoritmo.
- mu = coefficiente della lunghezza.
- $lambda1$ = coefficiente della forza interna.
- $lambda2$ = coefficiente della forza esterna.
- vi = coefficiente dell'area.
- $reverseColor$ = -1, 1 l'immagine segmentata può essere nera su sfondo bianco oppure bianca su sfondo nero.
- $method$ = metodo di segmentazione (chan, vector, multiphase).

oooooo

oooooooooooooooooooo

oo
ooo
ooo
o

Algoritmo di Chan Vese

Osservare le differenze tra la segmentazione di kass e quella di Chan-Vese

```
Istella = imread('immaginiTest/stella.png');  
chenvese(Istella,'small',300,0.2,1,1,0,1,'chan');  
Icerchi = imread('immaginiTest/dueCerchi.png');  
chenvese(Icerchi,'large',300,0.2,1,1,0,1,'chan');
```

Algoritmo di Chan Vese su immagini mediche

Osservare la differente segmentazione in funzione delle condizioni iniziale (maschera diversa)

```
Ipolmoni = imread('immaginiTest/polmoni.jpg');  
chenvese(Ipolmoni,'large',700,0.2,1,1,0,1,'chan');  
chenvese(Ipolmoni,'whole',700,0.2,1,1,0,1,'chan');  
  
Ineurone = imread('immaginiTest/neurone.jpg');  
chenvese(Ineurone,'whole',700,0.2,1,1,0,1,'chan');  
mask = zeros(size(Ineurone));  
mask(250:350, 300:400,:) = 1;  
chenvese(Ineurone,mask,1000,0.2,1,1,0,1,'chan');
```

Algoritmo di Chan Vese

Osservare il comportamento dell'algoritmo alla variazione di λ_1

Iforzalnterna = imread('immaginiTest/testParametroForzalnterna.png');

Con i parametri 'classici'

chenvese(Iforzalnterna,'medium',1000,0.2,1,1,0,1,'chan');

Variare λ_1

chenvese(Iforzalnterna,'medium',4000,0.2,**30**,1,0,1,'chan');

oooooo ooooooooooooooooooooo

oo
ooo
ooo
o

Algoritmo di Chan Vese

Osservare il comportamento dell'algoritmo alla variazione di λ_2

IforzaEsterna = imread('immaginiTest/testParametroForzaEsterna.png');

Con i parametri 'classici'

chenvese(IforzaEsterna,'medium',400,0.2,1,1,0,1,'chan');

Variare λ_2

mask = zeros(size(IforzaEsterna));

mask(331:719,371:711) = 1;

chenvese(IforzaEsterna,'medium',2500,0.2,1,**100**,0,1,'chan');

oooooo
oooooo

oooooooooooooooooooo
ooo

oo
ooo
ooo
o

Algoritmo di Chan Vese Vector

Osservare il risultato della segmentazione con il metodo 'vector'

```
Ilego4colori = imread('immaginiTest/lego4colori.jpg');
```

Algoritmo Chan-Vese

```
chenvese(Ilego4colori,'medium',800,0.2,1,1,0,1,'chan');
```

Algoritmo Chan-Vese Vector

```
chenvese(Ilego4colori,'medium',1000,0.2,1,1,0,1,'vector');
```

oooooo
oooooo
oooo

oo
ooo
ooo
o

Algoritmo di Chan Vese Vector

Osservare il risultato della segmentazione con il metodo 'vector' su immagini rumorose

```
Irosa = imread('immaginiTest/rosa.jpg');
```

```
Irosa = imresize(Irosa,0.2);
```

```
imshow(Irosa);
```

Applicare rumore

```
Irosa(:,:,1) = imnoise(Irosa(:,:,1),'speckle');
```

```
Irosa(:,:,2) = imnoise(Irosa(:,:,2),'salt & pepper',0.8);
```

```
chenvese(Irosa,'medium',800,0.2,1,1,0,1,'chan');
```

Usare 'vector'

```
chenvese(Irosa,'medium',200,0.2,1,1,0,1,'vector');
```

oooooo
oooooo
oooo

oo
ooo
ooo
o

Algoritmo di Chan Vese Multiphase

Osservare il risultato della segmentazione con il metodo 'multiphase'

```
Ilego4colori = imread('immaginiTest/lego4colori.jpg');
```

Algoritmo Chan-Vese Vector

```
chenvese(Ilego4colori,'medium',1000,0.2,1,1,0,1,'vector');
```

Algoritmo Chan-Vese Multiphase

```
chenvese(Ilego4colori,'medium',500,0.2,1,1,0,1,'multiphase');
```

```
Ilego5colori = imread('immaginiTest/lego5colori.jpg');
```

```
chenvese(Ilego5colori,'medium',1400,0.2,1,1,0,-1,'multiphase');
```